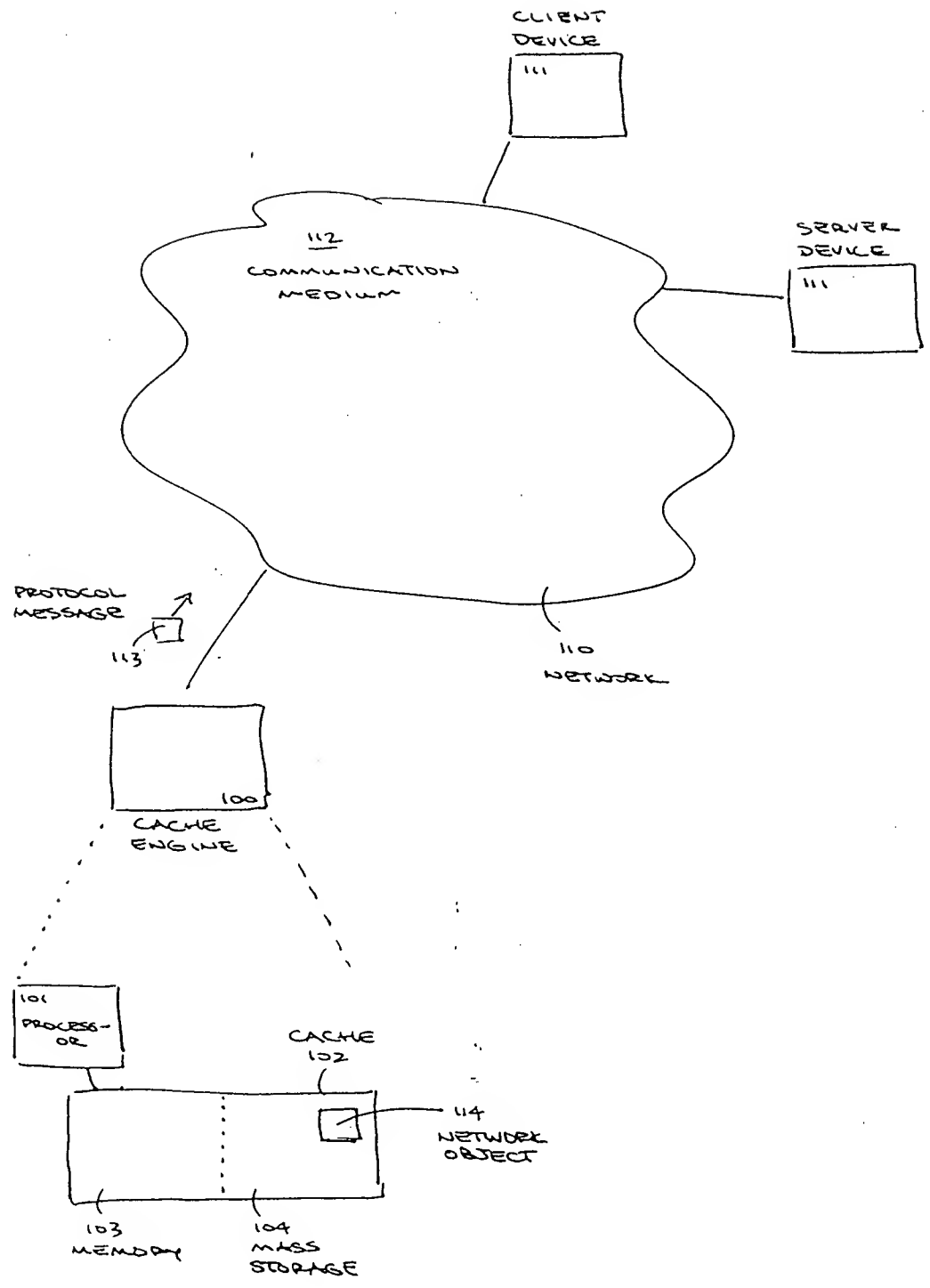


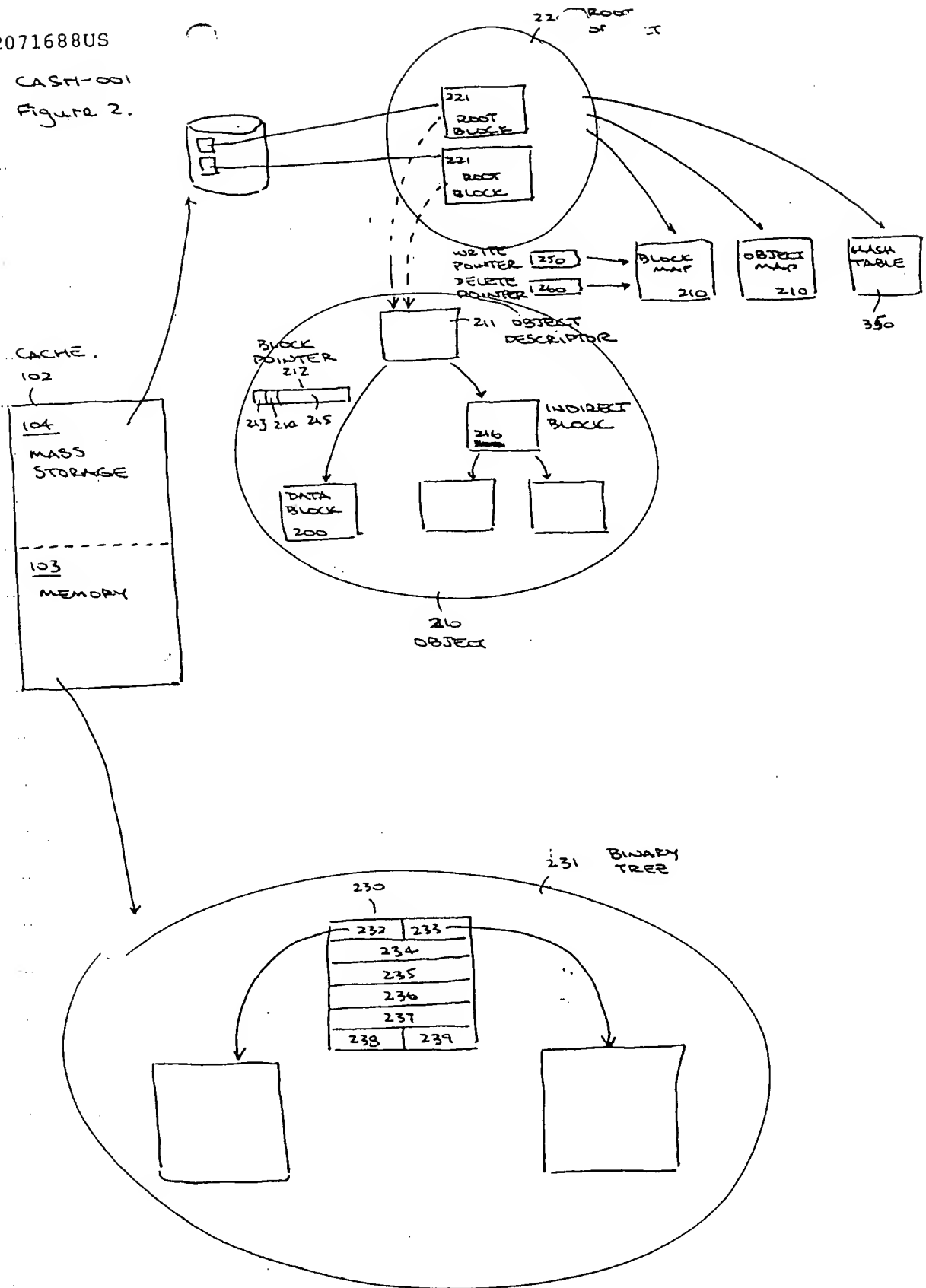
EM302071688US

CASH-001

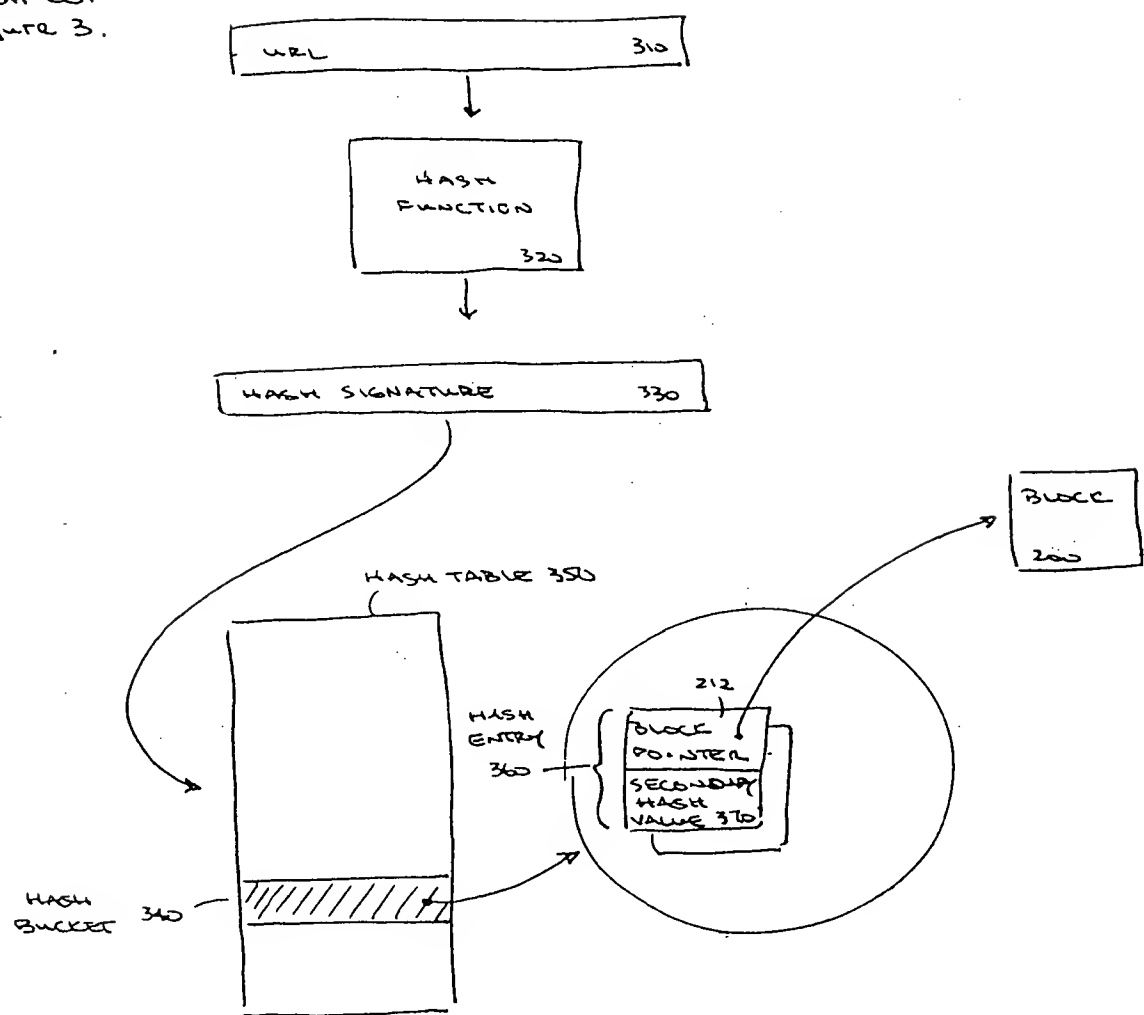
Figure 1.



CASH-001  
Figure 2.

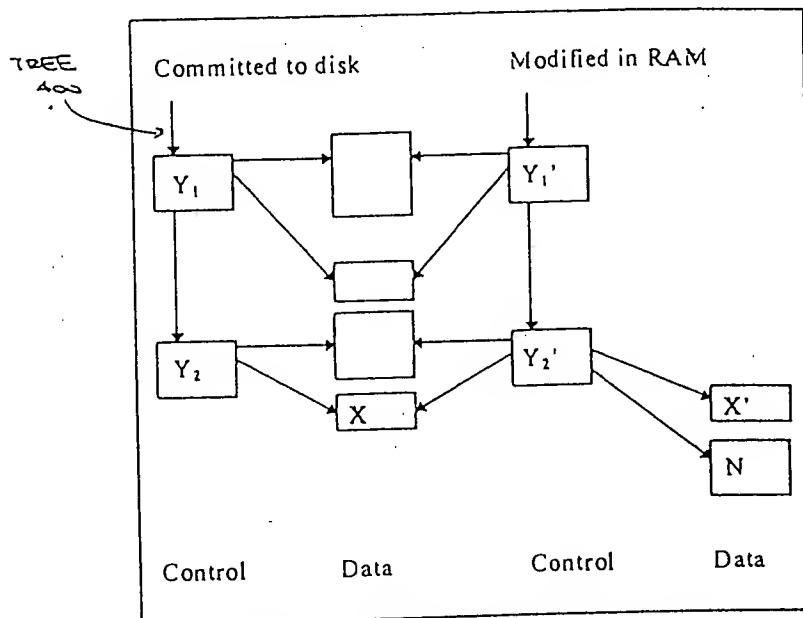


CASH-001  
Figure 3.



BRIS020/200000

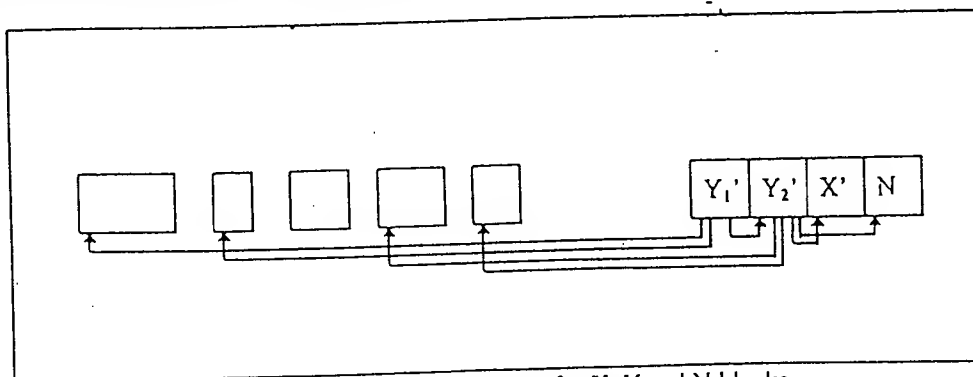
Figure 1 illustrates this bottom-up approach as it applies to the data and control blocks; the left half shows the committed information previously written to disk (of which parts may be in the RAM cache) and the right half shows the modified information held in the RAM cache allocated to previously free blocks and waiting to be written:



CASH-001  
FIGURE 4.

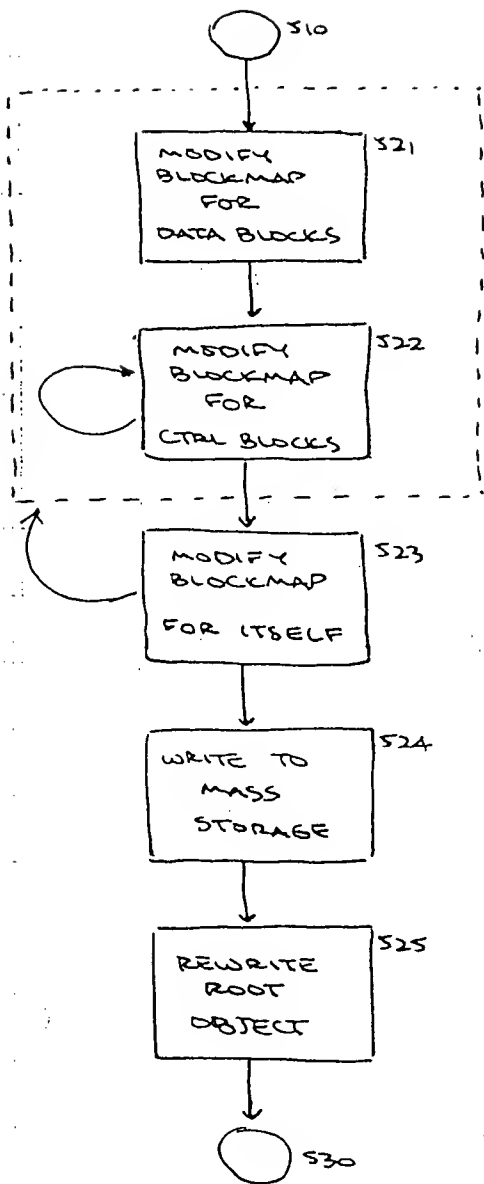
The N block represents a block containing new information. Note that the generic procedure performs a breadth first bottom up walk of the tree of modified control and data blocks. The result is that data blocks appear before control blocks and would ordinarily impose reverse seeks, from control blocks to data blocks, when the modified objects must be reread from the disk.

The Cache Engine improves on this by effectively walking the tree in a breadth first top down manner thus ensuring that control blocks are allocated to disk addresses ahead of other control blocks. The breadth first top down walk changes to a depth first walk at each control block which points to data blocks so that data blocks will be clustered together following the control block. This allows a simple read-ahead strategy applied to these control blocks to pre-fetch the appropriate data blocks. Figure 2 illustrates the layout of the modified and new blocks in our ongoing example:



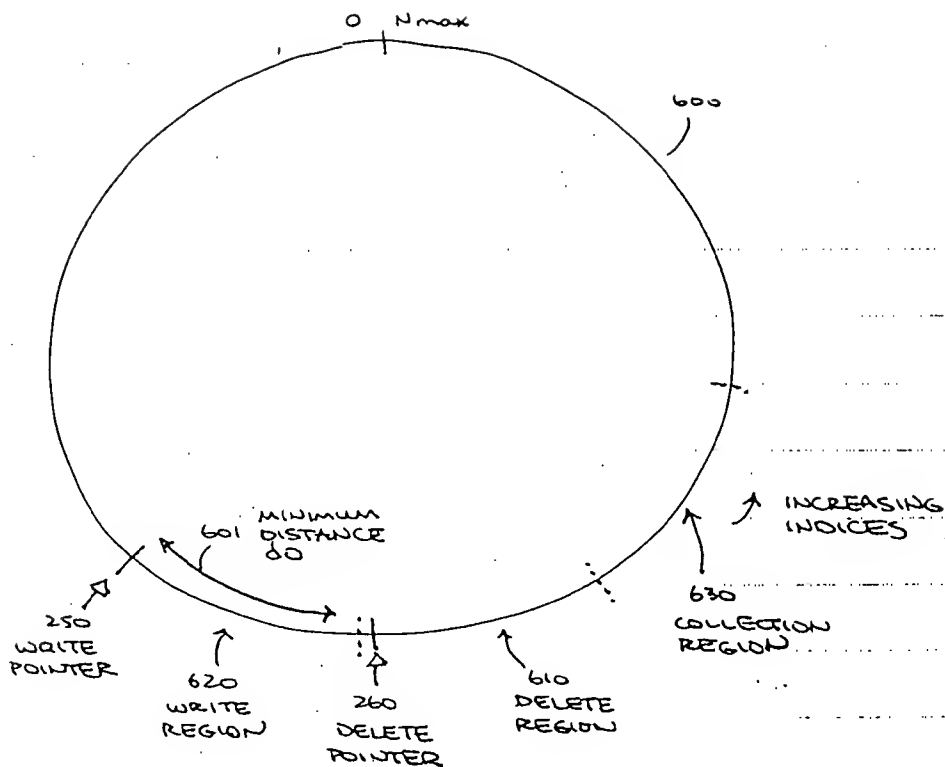
The Cache Engine further improves subsequent read performance.

BEST AVAILABLE COPY



**BEST AVAILABLE COPY**

[illegible]



090153 090153

BEST AVAILABLE COPY